Diffusion Recursive Least-Squares for Distributed Estimation Over Adaptive Networks

Federico S. Cattivelli, *Student Member, IEEE*, Cassio G. Lopes, *Student Member, IEEE*, and Ali. H. Sayed, *Fellow, IEEE*

***Abstract*—We study the problem of distributed estimation over adaptive networks where a collection of nodes are required to estimate in a collaborative manner some parameter of interest from their measurements. The centralized solution to the problem uses a fusion center, thus, requiring a large amount of energy for communication. Incremental strategies that obtain the global solution have been proposed, but they require the definition of a cycle through the network. We propose a diffusion recursive least-squares algorithm where nodes need to communicate only with their closest neighbors. The algorithm has no topology constraints, and requires no transmission or inversion of matrices, therefore saving in communications and complexity. We show that the algorithm is stable and analyze its performance comparing it to the centralized global solution. We also show how to select the combination weights optimally.**

***Index Terms*—Adaptive networks, consensus, cooperation, diffusion, distributed estimation, distributed processing.**

I. Introduction

A. Distributed Estimation

We study the problem of distributed estimation where a collection of nodes are required to estimate in a collaborative manner some parameter of interest from their measurements. In the centralized solution to the problem, measurements are transmitted to a central fusion center for processing, and the resulting estimate is communicated back to the nodes. This approach enables the calculation of the global solution, but has the disadvantage of requiring a large amount of energy and communication [2]. An alternative approach is the distributed solution, in which the nodes communicate only with their closest neighbors and processing is done locally at every node, thereby saving communications and network resources.

A distributed estimation approach should have the following desirable features.

- *Estimation performance:* The nodes in the network should obtain estimates that are close to the global solution.
- *Energy awareness and complexity:* The solution should minimize communications and local processing at the nodes.
- *Ad-hoc deployment:* The system should be able to cope with different, possibly dynamic, configurations of the network.

Distributed estimation algorithms have been proposed to address these issues to some extent. Depending on the manner by which the nodes communicate with each other, they may be referred to as incremental algorithms or diffusion algorithms. In the former, a cyclic path through the network is required, and nodes communicate with neighbors within this path. In the latter, nodes communicate with all of their neighbors, and no cyclic path is required.

In [3] and [4], a distributed incremental RLS solution was proposed for obtaining the exact global least-squares estimate. The algorithm requires the definition of a path through the network, which may not be practical for large networks or dynamic configurations. In [3]–[6], both incremental and diffusion LMS algorithms were proposed to perform distributed estimation. Distributed estimation algorithms of the consensus type were also proposed in [7]–[9]. The consensus implementations generally require two time scales: a slower time scale for the measurements and a faster time scale for processing iterations between measurements. This structure limits adaptation and the ability to track in real-time variations in the statistical properties of the data. One form of adaptivity was introduced in [10], where an isotropic diffusion algorithm based on least-squares was proposed. Nevertheless, the algorithm requires every node to transmit and invert a matrix at every iteration, which is generally prohibitive for large matrices or for low complexity sensor nodes.

We propose a distributed diffusion algorithm based on RLS that has performance close to the global solution and outperforms earlier solutions in terms of performance and complexity. The solution does not require transmission or inversion of matrices, therefore, saving in communications and computational complexity. The algorithm has no topological constraints, and functions as a fully adaptive, recursive and distributed solution that is asymptotically unbiased and stable under the modeling assumptions of Section IV.

A key contribution of this work is not only to propose and derive a diffusion least-squares solution, but to also study its mean-square performance in some detail. This latter task is challenging due to the fact that nodes in every neighborhood interact with each other and, therefore, a successful analysis must take into account both the temporal and spatial interconnectedness of the data.

Manuscript received June 4, 2007; revised October 1, 2007. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Alper Tunga Erdogan. This work was supported in part by the National Science Foundation under awards ECS-0601266 and ECS-0725441. An earlier version of this work appeared in the *Proceedings of the IEEE Workshop on Signal Processing Advances in Wireless Communications* (SPAWC), Helsinki, Finland, June 2007, pp. 1–5.

The authors are with the Department of Electrical Engineering, University of California, Los Angeles, CA 90095 USA (e-mail: fcattiv@ee.ucla.edu; cassio@ee.ucla.edu; sayed@ee.ucla.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2007.913164

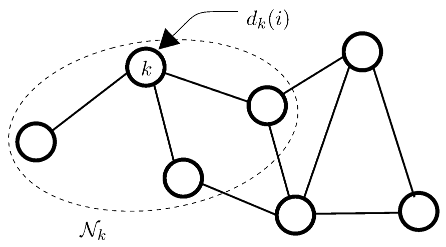

Fig. 1. Network showing a neighborhood $\mathcal{N}_k$ of node k. At time i, node k collects a measurement $d_k(i)$.

B. Adaptive Networks

We adopt the term adaptive networks from [3], [4] to refer to a collection of nodes that interact with each other, and function as a single adaptive entity that is able to respond to data in real-time and also track variations in their statistical properties. A set of nodes that collaborate to estimate a parameter of interest using the diffusion RLS algorithm derived in this paper is an example of such an adaptive network.

The estimation problem is formally defined in Section II, followed by the specification of the diffusion RLS algorithm in Section III. We analyze the performance of the algorithm in terms of its mean and mean-square behavior in Section IV. In Section V we discuss some common choices of combiner coefficients, including an optimal choice. Simulation results are presented in Section VI.

II. The Estimation Problem

A. Global Least-Squares Problem

To begin with, consider a set of N nodes spatially distributed over some region as in Fig. 1. Let $\mathcal{N}_k$ denote the *closed* neighborhood of node k (i.e., the set of all neighbors of node k including itself). The objective of the network of nodes is to collectively estimate an unknown deterministic column vector of length M, denoted by w^o, using least-squares estimation. At *every* time instant i, node k collects a measurement $d_k(i)$ that is assumed to be related to the unknown vector by

$$d_k(i) = u_{k,i} w^o + v_k(i) \tag{1}$$

where $u_{k,i}$ is a row vector of length M (the regressor of node k at time i), and $v_k(i)$ is a zero-mean, spatially uncorrelated Gaussian white noise process with variance $\sigma_{v_k}^2$ and independent of $u_{k,i}$. Linear models of the form (1) are able to capture or approximate well many input–output relations for estimation purposes [11 p. 90].

At time i, we collect the measurements and noise samples of all N nodes into vectors y_i and v_i of length N, and the regressors into an N by M matrix H_i, as follows:

$$\begin{aligned} y_i &= \text{col}\{d_1(i), \ldots, d_N(i)\} \quad (N \times 1) \\ H_i &= \text{col}\{u_{1,i}, \ldots, u_{N,i}\} \quad (N \times M) \\ v_i &= \text{col}\{v_1(i), \ldots, v_N(i)\} (N \times 1). \end{aligned}$$

Let v_i^* denote the complex conjugate transpose of vector v_i. The covariance matrix of the noise vector is

$$R_v = \mathrm{E} v_i v_i^* = \text{diag}\left\{\sigma_{v_1}^2, \ldots, \sigma_{v_N}^2\right\} (N \times N).$$

We further collect the regressors, measurements and covariance matrices from time 0 up to time i as follows:

$$\begin{aligned} \mathcal{Y}_i &= \text{col}\{y_i, \ldots, y_0\} \\ \mathcal{H}_i &= \text{col}\{H_i, \ldots, H_0\} \\ \mathcal{V}_i &= \text{col}\{v_i, \ldots, v_0\} \end{aligned}$$

and let $\mathcal{R}_{v,i} = \mathrm{E}\mathcal{V}_i \mathcal{V}_i^*$.

The objective is to estimate w^o by solving the following weighted, regularized, least-squares problem

$$\min_w \|w - \bar{w}\|_{\Pi_i}^2 + \|\mathcal{Y}_i - \mathcal{H}_i w\|_{\mathcal{W}_i}^2 \tag{2}$$

The solution w_i is given by [11]

$$w_i = \bar{w} + \left[\Pi_i + \mathcal{H}_i^* \mathcal{W}_i \mathcal{H}_i\right]^{-1} \mathcal{H}_i^* \mathcal{W}_i (\mathcal{Y}_i - \mathcal{H}_i \bar{w}) \tag{3}$$

where $\Pi_i > 0$ is a regularization matrix and $\mathcal{W}_i \geq 0$ is a weighting matrix. Both Π_i and $\mathcal{W}_i$ are Hermitian.

An exponentially weighted version of (2) can be formulated by choosing

$$\mathcal{W}_i = \mathcal{R}_{v,i}^{-1} \Lambda_i \qquad \Pi_i = \lambda^{i+1} \Pi$$

with $0 < \lambda \leq 1, \Pi > 0$ and

$$\Lambda_i \overset{\triangle}{=} \text{diag}\left\{I_N, \lambda I_N, \ldots, \lambda^i I_N\right\}.$$

Usually, $\Pi = \delta^{-1} I_M$ where $\delta > 0$ is large. Often, in least-squares estimation, when the noise variances $\{\sigma_{v_k}^2\}$ are unknown, the weighting matrix $\mathcal{W}_i$ is simply replaced by $\mathcal{W}_i = \Lambda_i$. The subsequent arguments and derivations in Sections III and IV still hold with $\sigma_{v_k}^2$ replaced by unity [for example, in the statement of the diffusion RLS algorithm (12)]. The $\{\sigma_{v_k}^2\}$ may also be interpreted as some scalar weights.

For the choice $\bar{w} = 0$, the estimation problem (2) becomes

$$w_i = \arg\min_w \left\{ \lambda^{i+1} \|w\|_\Pi^2 + \sum_{j=0}^{i} \lambda^{i-j} \sum_{l=1}^{N} \frac{|d_l(j) - u_{l,j} w|^2}{\sigma_{v_l}^2} \right\}. \tag{4}$$

We refer to this problem as the *global least-squares problem*, since at time i, the solution takes into account all measurements from all nodes up to time i. This solution may be computed by using a centralized approach, or the distributed incremental RLS algorithm of [3], [4].

B. Clustered Least-Squares Problem

We now proceed to propose distributed estimation schemes where nodes have access to limited data, namely, the data from the neighboring nodes. Thus, when node k can only share measurements and regressors with its neighbors, it can locally solve the following least-squares problem:

$$\psi_{k,i} = \arg\min_{w} \left\{ \lambda^{i+1}\|w\|_{\Pi}^2 + \sum_{j=0}^{i} \lambda^{i-j} \sum_{l=1}^{N} \frac{c_{l,k}|d_l(j) - u_{l,j}w|^2}{\sigma_{v_l}^2} \right\} \quad (5)$$

for some choice of positive weighting coefficients $c_{l,k}$ such that

$$c_{l,k} = 0 \,\text{if}\, l \notin \mathcal{N}_k$$

Let C denote an $N \times N$ matrix such that its $\{l,k\}$ element is $c_{l,k}$. For reasons that will become clearer later in this paper, we also choose the coefficients $c_{l,k}$ such that

$$\mathbb{1}^T C = \mathbb{1}^T \,\text{and}\, C\mathbb{1} = \mathbb{1}$$

where $\mathbb{1}$ represents an $N \times 1$ vector whose entries are all unity. Node k weights the data obtained by node l using the coefficient $c_{l,k}$. A zero $c_{l,k}$ indicates that nodes l and k are not connected (i.e., do not interact). A nonzero $c_{l,k}$ allows node k to give some reliability/relevance measure to the data obtained from node l.

Note that the nodes in the network will generally have access to different data, so their estimates of w^o will be solutions to different least-squares problems. Naturally, we want these estimates to be close to the global least-squares solution of (4).

The coefficients $c_{l,k}$ can be incorporated into the weighting matrix of (2) by replacing $\mathcal{W}_i$ with

$$\mathcal{W}_{k,i} = \mathcal{R}_{v,i}^{-1}\Lambda_i \text{diag}\{C_k, C_k, \ldots, C_k\} \quad (6)$$

where $C_k = \text{diag}\{Ce_k\}$ of size $N \times N$, and e_k is the $N \times 1$ vector with a unity entry in position k and zeros elsewhere.

Now, for every node k, we express its local estimate $\psi_{k,i}$ resulting from (5) as a perturbation of w^o, say for some error vectors $z_{k,i}$

$$\begin{bmatrix} \psi_{1,i} \\ \psi_{2,i} \\ \vdots \\ \psi_{N,i} \end{bmatrix} = \begin{bmatrix} I_M \\ I_M \\ \vdots \\ I_M \end{bmatrix} w^o + \begin{bmatrix} z_{1,i} \\ z_{2,i} \\ \vdots \\ z_{N,i} \end{bmatrix}. \quad (7)$$

Note that this model holds exactly for any least-squares problem of the form (7) when there is no regularization ($\Pi = 0$), since in this case the solution $\psi_{k,i}$ from (3) is

$$\psi_{k,i} = [\mathcal{H}_i^* \mathcal{W}_{k,i} \mathcal{H}_i]^{-1} \mathcal{H}_i^* \mathcal{W}_{k,i} \mathcal{Y}_i.$$

Using

$$\mathcal{Y}_i = \mathcal{H}_i w^o + \mathcal{V}_i$$

we find that (7) holds with

$$z_{k,i} = [\mathcal{H}_i^* \mathcal{W}_{k,i} \mathcal{H}_i]^{-1} \mathcal{H}_i^* \mathcal{W}_{k,i} \mathcal{V}_i.$$

Note that $z_{k,i}$ is zero-mean and Gaussian as well. By the same token, model (7) is a good approximation when $\Pi = \delta^{-1}I_M$ and δ is large enough.

Given the local estimates $\psi_{k,i}$ and (7) we now investigate the following least-squares problem:

$$\min_{w} \left\| \begin{bmatrix} \psi_{1,i} \\ \psi_{2,i} \\ \vdots \\ \psi_{N,i} \end{bmatrix} - \begin{bmatrix} I_M \\ I_M \\ \vdots \\ I_M \end{bmatrix} w \right\|_{\Gamma}^2 \quad (8)$$

with a weighting matrix $\Gamma > 0$. If we assume that the individual solutions $\psi_{k,i}$ are good approximations for the optimal solution w^o, then we would expect the solution for problem (8) to be closer to the optimal than the individual $\psi_{k,i}$ for a proper choice of Γ. For instance, for the choice of weighting matrix (assumed positive-definite, which holds for full-rank data matrices $\mathcal{H}_i$)

$$\Gamma = \begin{bmatrix} \mathcal{H}_i^* \mathcal{W}_{1,i} \mathcal{H}_i & & \\ & \ddots & \\ & & \mathcal{H}_i^* \mathcal{W}_{N,i} \mathcal{H}_i \end{bmatrix} \quad (9)$$

we have that the solution of (8) is

$$w_i = \left[\mathcal{H}_i^* \left(\sum_{k=1}^{N} \mathcal{W}_{k,i} \right) \mathcal{H}_i \right]^{-1} \mathcal{H}_i^* \left(\sum_{k=1}^{N} \mathcal{W}_{k,i} \right) \mathcal{Y}_i. \quad (10)$$

For the choice of coefficients $c_{l,k}$ from before, which satisfy $C\mathbb{1} = \mathbb{1}$, it follows that:

$$\sum_{k=1}^{N} C_k = I_N.$$

Thus (10) becomes

$$w_i = \left[\mathcal{H}_i^* \mathcal{R}_{v,i}^{-1} \Lambda_i \mathcal{H}_i \right]^{-1} \mathcal{H}_i^* \mathcal{R}_{v,i}^{-1} \Lambda_i \mathcal{Y}_i$$

which is precisely the solution of (4) (i.e., the global solution) when there is no regularization.

This observation suggests an estimation process in two steps. First, every node solves a *local* least-squares problem using local data as in (5), and second, the nodes communicate to solve problem (8). The inconvenience of this approach is that the second step requires knowledge of the local estimate of every other node, and also knowledge of all regressors in the network. That is, the second step requires global communication and sharing of data. We are instead interested in a distributed solution that relies *solely* on local interactions. Motivated by this discussion we now propose a modification whereby nodes will only need to share data within their neighborhood.

To do so, consider the case where we replace the weighting matrix Γ in (8) and (9) by a node-dependent diagonal matrix of the form

$$\Gamma_k = \text{diag}\{a_{1,k}I, \ldots, a_{N,k}I\}$$

where $a_{l,k} \geq 0 \in \mathbb{R}$ and $a_{l,k} = 0$ if $l \notin \mathcal{N}_k$. Let A denote the $N \times N$ matrix whose $\{l,k\}$ entry is $a_{l,k}$. We choose the coefficients $a_{l,k}$ such that

$$\mathbb{1}^T A = \mathbb{1}^T$$

Again, just like the coefficients $c_{l,k}$, a zero $a_{l,k}$ indicates that nodes l and k are not connected. A nonzero $a_{l,k}$ allows node k to give some reliability/relevance measure to the local estimate

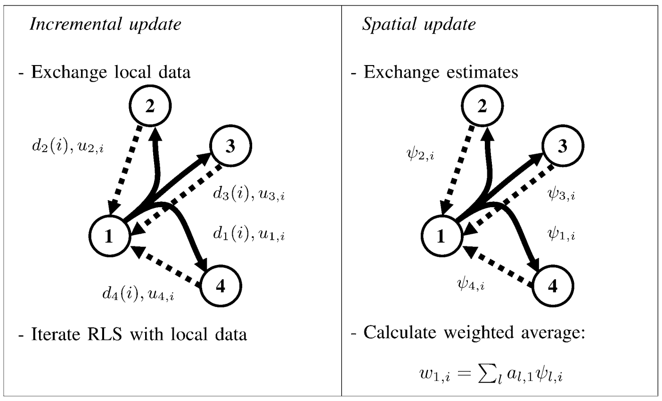

Fig. 2. Diffusion RLS algorithm at node 1. Incremental update: all neighboring nodes exchange measurements and regressors and compute $\psi_{k,i}$. Spatial update: neighboring nodes exchange $\psi_{k,i}$ and perform a weighted average to obtain $w_{k,i}$.

obtained by node l. We, thus, employ two sets of weights: $\{c_{l,k}\}$ for local processing of data and $\{a_{l,k}\}$ for local processing of weight estimates.

Then, the solution to the least-squares problem (8) at node k takes the form

$$w_{k,i} = \sum_{l=1}^{N} a_{l,k}\psi_{l,i}. \tag{11}$$

This means that the second step of the aforementioned method is replaced by a weighted average of the local estimates in the neighborhood of node k. Convex combinations of the estimates of adaptive filters as in (11) have been studied in [12]–[14].

This sequence of two least-squares problems represents an attempt to solve the global least-squares problem (4) in a distributed manner. However, the method is still not adaptive, since we first need to calculate local estimates, and then average the results as in (11). We can make the method adaptive if we perform both steps simultaneously for every measurement in real time. That is, first aggregate the new data into the local estimate, and then combine estimates with neighbors. This procedure, referred to as diffusion RLS, is described in the following section, and has good convergence properties as shown in Section IV.

III. The Diffusion RLS Algorithm

We, therefore, propose a diffusion RLS algorithm to collectively estimate w^o from individual measurements in two steps as follows.

1) At time i, the nodes communicate their measurements $d_k(i)$ and regressors $u_{k,i}$ with their neighbors, and use this data to update their local estimates using RLS iterations (via a so-called incremental update). The resulting preestimates are named $\psi_{k,i}$ as in (5).
2) The nodes communicate their local preestimates with their neighbors and perform a weighted average as in (11) to obtain the estimate $w_{k,i}$ (via a so-called spatial update).

The algorithm, shown schematically in Fig. 2, is described by (12). A derivation of (12) can be found in Appendix A. It starts by selecting matrices $A \in \mathbb{R}^{N\times N}$ and $C \in \mathbb{R}^{N\times N}$ with nonnegative entries such that $a_{l,k} = c_{l,k} = 0$ if $l \notin \mathcal{N}_k$, $\mathbb{1}^T A = \mathbb{1}^T$, $\mathbb{1}^T C = \mathbb{1}^T$ and $C\mathbb{1} = \mathbb{1}$.

Diffusion RLS Algorithm

```
Start with $w_{k,-1} = 0$ and $P_{k,-1} = \Pi^{-1}$ for each node $k$
For every time instant $i$, repeat
    **Incremental update**: for every node $k$, repeat
        $\psi_{k,i} = w_{k,i-1}$
        $P_{k,i} = \lambda^{-1}P_{k,i-1}$
        for all $l \in \mathcal{N}_k$
            $\psi_{k,i} \leftarrow \psi_{k,i} + \frac{c_{l,k}P_{k,i}u_{l,i}^*[d_l(i) - u_{l,i}\psi_{k,i}]}{\sigma_{v_l}^2 + c_{l,k}u_{l,i}P_{k,i}u_{l,i}^*}$
            $P_{k,i} \leftarrow P_{k,i} - \frac{c_{l,k}P_{k,i}u_{l,i}^*u_{l,i}P_{k,i}}{\sigma_{v_l}^2 + c_{l,k}u_{l,i}P_{k,i}u_{l,i}^*}$
        end
    **Spatial update:** for every node $k$, repeat
        $w_{k,i} = \sum_{l\in\mathcal{N}_k} a_{l,k}\psi_{l,i}.$        (12)
```

Note that the algorithm requires no matrix inversion, and nodes only need to know the estimates $\psi_{k,i}$, measurements $d_k(i)$ and regressors $u_{k,i}$ of their neighbors. Thus, for every new measurement, every node needs to communicate a total of $2M+1$ scalars to neighboring nodes. For comparison purposes, the incremental RLS algorithm of [3], [4] and the diffusion algorithm of [10] require that, for every measurement, every node transmit a vector of size M and a Hermitian matrix of size $M \times M$, requiring a total communication of $(M^2 + 3M)/2$ scalars. In Section VI we compare these algorithms in terms of their mean-square performance as a function of communicated scalars.

IV. Analysis

In this section, we analyze the performance of algorithm (12) and show that it is asymptotically unbiased in the mean and converges in the mean-square error sense under some simplifying assumptions. We also provide an expression for its steady-state excess mean-squared-error (EMSE) and mean-square deviation (MSD) and compare its performance to the global least-squares solution (4). The main results are the detailed expressions (21)–(23) and (29)–(30) for the MSD and EMSE performance of the adaptive network.

A. Data Model

As is well known, it is rather challenging to study the mean-square performance of single stand-alone adaptive filters [11], [15]–[18]. Several simplifying assumptions have been traditionally adopted in the literature to gain insight into the performance of such adaptive algorithms. The challenges are compounded in the adaptive network case because we now face a dynamic and interconnected collection of nodes that influence each other's behavior. To proceed with the analysis we shall therefore introduce similar assumptions to what has been used before in the adaptive literature, and use them to derive useful performance measures. Simulations show that the results obtained in this manner match well with real performance for forgetting factors λ sufficiently close to unity and for data whose statistical properties do not vary fast with time.

We start with the following assumption on the regressors.

Assumption 1: The regressors $u_{k,i}$ are zero-mean and temporally independent. Moreover, the covariance matrix $R_{u_k} = \mathrm{E}u_{k,i}^* u_{k,i}$ is invariant over time.

From (12) we can observe that after the incremental update is complete, $P_{k,i}$ is obtained from a series of rank-one updates to $\lambda^{-1}P_{k,i-1}$. Specifically, it holds that

$$\begin{aligned}P_{k,i}^{-1} =& \lambda P_{k,i-1}^{-1} + \sum_{l=1}^{N} \frac{c_{l,k}}{\sigma_{v_l}^2} u_{l,i}^* u_{l,i} \\ =& \lambda^{i+1}\Pi + \sum_{j=0}^{i} \lambda^{i-j} \sum_{l=1}^{N} \frac{c_{l,k}}{\sigma_{v_l}^2} u_{l,j}^* u_{l,j}. \end{aligned} \qquad (13)$$

We are interested in the steady-state behavior of the matrix $P_{k,i}$. As $i \to \infty$, and for $0 \ll \lambda < 1$, the steady-state mean value of $P_{k,i}^{-1}$ is given by

$$\lim_{i\to\infty} \mathrm{E}P_{k,i}^{-1} = \frac{1}{1-\lambda} \sum_{l=1}^{N} \frac{c_{l,k}}{\sigma_{v_l}^2} R_{u_l} \triangleq P_k^{-1}. \qquad (14)$$

In order to make the performance analysis tractable, we introduce the following ergodicity assumption.

Assumption 2: $\exists i_0$ such that for all $i > i_0, P_{k,i}^{-1}$ can be replaced by $\mathrm{E}P_{k,i}^{-1} = P_k^{-1}$.

Assumption 2: states that the expected value of a random process can be replaced by its time average. This is a common assumption in the analysis of the performance of RLS-type algorithms (see, for example, [11, pp. 318–319]), and yields good results in practice as is illustrated by simulation in Section VI. Furthermore, as is also common in the analysis of the performance of RLS-type algorithms, to study mean-square convergence we will replace the random matrix $P_{k,i}$ by P_k for i large enough.

B. Mean Performance

Combining the data model (1) with (12), we have the following relation for the local estimate after the incremental update is complete:

$$\begin{aligned}P_{l,i}^{-1}\psi_{l,i} =& \lambda P_{l,i-1}^{-1} w_{l,i-1} + \sum_{m=1}^{N} \frac{c_{m,l}}{\sigma_{v_m}^2} u_{m,i}^* d_m(i) \\ =& \lambda P_{l,i-1}^{-1} w_{l,i-1} + \sum_{m=1}^{N} \frac{c_{m,l}}{\sigma_{v_m}^2} u_{m,i}^* \left[u_{m,i} w^o + v_m(i)\right]. \end{aligned}$$

By using (13) in the previous expression we arrive at

$$P_{l,i}^{-1}\psi_{l,i} = \lambda P_{l,i-1}^{-1}\tilde{w}_{l,i-1} + P_{l,i}^{-1} w^o + \sum_{m=1}^{N} \frac{c_{m,l}}{\sigma_{v_m}^2} u_{m,i}^* v_m(i)$$

where $\tilde{w}_{k,i} \triangleq w_{k,i} - w^o$. It then follows that the error vector $\tilde{w}_{k,i}$ satisfies:

$$\begin{aligned}\tilde{w}_{k,i} &= \sum_{l=1}^{N} a_{l,k}\left[\psi_{l,i} - w^o\right] \\ &= \sum_{l=1}^{N} a_{l,k} P_{l,i}\left[\lambda P_{l,i-1}^{-1}\tilde{w}_{l,i-1} + \sum_{m=1}^{N} \frac{c_{m,l}}{\sigma_{v_m}^2} u_{m,i}^* v_m(i)\right]. \end{aligned} \qquad (15)$$

Taking expectations of both sides yields the following result:

$$\mathrm{E}\{\tilde{w}_{k,i}\} = \lambda \mathrm{E}\left\{\sum_{l=1}^{N} a_{l,k} P_{l,i} P_{l,i-1}^{-1} \tilde{w}_{l,i-1}\right\}.$$

We group the vectors $\tilde{w}_{k,i}$ into a matrix

$$\tilde{W}_i \triangleq \mathrm{row}\{\tilde{w}_{1,i}, \ldots, \tilde{w}_{N,i}\} \quad (M \times N). \qquad (16)$$

We also define the $N \times N$ matrix $A = [a_{l,k}]$. Using assumption 2 and noting that $P_{k,i}^{-1}$ becomes independent of i for $i > i_0$, we have for large enough i

$$\begin{aligned}\mathrm{E}\{\tilde{W}_i\} &= \lambda \mathrm{E}\{\tilde{W}_{i-1}\}A \\ &= \lambda^{i-i_0}\mathrm{E}\{\tilde{W}_{i_0}\}A^{i-i_0}. \end{aligned}$$

Assuming that all elements of $\mathrm{E}\{\tilde{W}_{i_0}\}$ are bounded in absolute value by some finite constant a, and since all elements of A^i are between zero and one, we have that every element of $\mathrm{E}\{\tilde{W}_i\}$ is bounded in absolute value by $\lambda^{i-i_0}Na$. Thus, for $0 \ll \lambda < 1$, every element of $\mathrm{E}\{\tilde{W}_i\}$ converges to zero as $i \to \infty$, and the estimator is asymptotically unbiased.

C. Mean-Square Performance

We now show that the algorithm converges in the mean-square sense, i.e., $\mathrm{E}\|\tilde{w}_{k,i}\|^2 < \infty$ as $i \to \infty$, and derive expressions for the steady-state excess mean-square error (EMSE) and steady-state mean-square deviation (MSD). These are defined, at every node k, as [11]

$$\begin{aligned}\mathrm{MSD}_k &\triangleq \lim_{i\to\infty} \mathrm{E}\|\tilde{w}_{k,i}\|^2 \\ \mathrm{EMSE}_k &\triangleq \lim_{i\to\infty} \mathrm{E}\,|\,u_{k,i}\tilde{w}_{k,i-1}|^2. \end{aligned}$$

The mean-square error is given by

$$\mathrm{MSE}_k \triangleq \lim_{i\to\infty} \mathrm{E}|d_k(i) - u_{k,i}w_{k,i-1}|^2 = \sigma_{v_k}^2 + \mathrm{EMSE}_k.$$

We begin with the general case, where we only use assumptions 1 and 2. Subsequently, we specialize the general result to some interesting special cases, namely the case where the noise variances and regressor covariance matrices are the same for every node (spatial invariance).

1) General Case: Starting from (15) and assumption 2 we have for large enough i

$$\begin{aligned}\tilde{w}_{k,i} &= \lambda \sum_{l=1}^{N} a_{l,k}\tilde{w}_{l,i-1} + \sum_{l=1}^{N} a_{l,k}P_l \sum_{m=1}^{N} \frac{c_{m,l}}{\sigma_{v_m}^2} u_{m,i}^* v_m(i) \\ &= \lambda \tilde{W}_{i-1}Ae_k + \sum_{l=1}^{N} a_{l,k}P_l H_i^* C_l R_v^{-1} v_i\end{aligned}$$

where $\tilde{W}_i$ was defined in (16) and, as before, we are using the notation $C_l = \mathrm{diag}\{Ce_l\}$. We can represent $\tilde{w}_{k,i}$ using a more compact notation as follows:

$$\tilde{w}_{k,i} = \lambda \tilde{W}_{i-1}Ae_k + \tilde{P}\tilde{B}_i Ae_k$$

where we introduced the matrices $\tilde{P}$ of size $M \times MN$ and $\tilde{B}_i$ of size $MN \times N$

$$\begin{aligned}\tilde{P} &\triangleq [P_1 P_2 \cdots P_N] \\ \tilde{B}_i &\triangleq \begin{bmatrix} H_i^* C_1 R_v^{-1} v_i & & & \\ & H_i^* C_2 R_v^{-1} v_i & & \\ & & \ddots & \\ & & & H_i^* C_N R_v^{-1} v_i \end{bmatrix} \\ &= \begin{bmatrix} H_i^* C_1 R_v^{-1} & & & \\ & H_i^* C_2 R_v^{-1} & & \\ & & \ddots & \\ & & & H_i^* C_N R_v^{-1} \end{bmatrix} (I_N \otimes v_i)\end{aligned}$$

and $\otimes$ denotes the Kronecker product of two matrices. It follows that

$$\tilde{W}_i = \lambda \tilde{W}_{i-1}A + \tilde{P}\tilde{B}_i A$$

for large i. That is

$$\tilde{W}_i = \tilde{W}_{i_0}(\lambda A)^{i-i_0} + \tilde{P} \sum_{j=i_0+1}^{i} \tilde{B}_j A(\lambda A)^{i-j}$$

and the kth column of $\tilde{W}_i$ is

$$\tilde{w}_{k,i} = \tilde{W}_{i_0}(\lambda A)^{i-i_0}e_k + \tilde{P} \sum_{j=i_0+1}^{i} \tilde{B}_j A(\lambda A)^{i-j}e_k. \quad (17)$$

We now consider the mean-square deviation, $\mathrm{E}\|\tilde{w}_{k,i}\|^2$ or $\mathrm{E}\ \mathrm{Tr}(\tilde{w}_{k,i}\tilde{w}_{k,i}^*)$ as $i \to \infty$. When we form the product $\tilde{w}_{k,i}\tilde{w}_{k,i}^*$, the cross terms in (17) vanish under expectation, since $\mathrm{E}v_i = 0$. The euclidean norm of the first term is given by

$$\mathrm{E}\|\tilde{W}_{i_0}(\lambda A)^{i-i_0}e_k\|^2 = \lambda^{2(i-i_0)}\mathrm{E}\left\|\sum_{l=1}^{N} \tilde{w}_{l,i_0}[A^{i-i_0}]_{l,k}\right\|^2.$$

Noting that A^{i-i_0} has nonnegative entries and satisfies $\mathbb{1}^T A^{i-i_0} = \mathbb{1}^T$, we can see that the above expression vanishes for large i. We are, therefore, left with the outer product of the second term of (17) with its conjugate. We first define the matrix

$$J^{j,l} \triangleq A(\lambda A)^{i-j}e_k e_k^T(\lambda A^T)^{i-l}A^T \qquad (N \times N). \quad (18)$$

Then we have

$$\begin{aligned}&\mathrm{E}(I_N \otimes v_j)J^{j,l}(I_N \otimes v_l)^* \\ &\qquad = \mathrm{E}(I_N \otimes v_j)(J^{j,l} \otimes 1)(I_N \otimes v_l^*).\end{aligned}$$

Using the property $(A \otimes B)(C \otimes D) = AC \otimes BD$, we arrive at

$$\mathrm{E}(I_N \otimes v_j)J^{j,l}(I_N \otimes v_l)^* = J^{j,l} \otimes \mathrm{E}(v_j v_l^*) = \delta_{j,l}(J^{j,j} \otimes R_v)$$

where $\delta_{j,l}$ denotes the Kronecker delta. For simplicity of notation, we will denote the matrix $J^{j,j}$ by J^j, and $[J^j]_{m,l}$ will denote the $\{m, l\}$ element of J^j. We now define the matrix

$$K_j \triangleq \tilde{B}_j A(\lambda A)^{i-j}e_k e_k^T(\lambda A^T)^{i-j}A^T\tilde{B}_j^* = \tilde{B}_j J^j \tilde{B}_j^*.$$

Note that K_j is a block matrix consisting of N blocks by N blocks of size $M \times M$ each, and the $\{m, l\}$ block is given by

$$K_{j,m,l} = [J^j]_{m,l}H_j^* C_m R_v^{-1} C_l H_j.$$

Now, after taking expectations

$$\mathrm{E}\{K_{j,m,l}\} = [J^j]_{m,l} \sum_{n=1}^{N} \frac{R_{u_n} c_{n,m} c_{n,l}}{\sigma_{v_n}^2}. \quad (19)$$

We, thus, have

$$\mathrm{E}\|\tilde{w}_{k,i}\|^2 = \mathrm{ETr}\left[\tilde{P}\left(\sum_{j=i_0+1}^{i} K_j\right)\tilde{P}^*\right].$$

From the definition of $\tilde{P}$, and noting that P_l is Hermitian for $l = 1 \cdots N$, we get

$$\mathrm{E}\|\tilde{w}_{k,i}\|^2 = \mathrm{Tr}\left(\sum_{j=i_0+1}^{i}\sum_{l=1}^{N}\sum_{m=1}^{N} P_m \mathrm{E}\{K_{j,m,l}\}P_l\right). \quad (20)$$

By inserting (19) and (18) into (20) we obtain

$$\begin{aligned}\mathrm{E}\|\tilde{w}_{k,i}\|^2 = &\sum_{j=i_0+1}^{i}\sum_{l=1}^{N}\sum_{m=1}^{N}\sum_{n=1}^{N} \\ &\left\{\mathrm{Tr}(P_m R_{u_n} P_l)\frac{c_{n,m}c_{n,l}}{\sigma_{v_n}^2}\lambda^{2(i-j)}\right. \\ &\left.\times [A^{i-j+1}]_{m,k}[(A^T)^{i-j+1}]_{k,l}\right\}\end{aligned}$$

Finally, the mean-square deviation at node k for the diffusion RLS algorithm (12) is given by

$$\boxed{\begin{aligned}\mathrm{MSD}_k^{\mathrm{diff}} = \lim_{i\to\infty}\sum_{j=0}^{i}\sum_{l=1}^{N}\sum_{m=1}^{N}\sum_{n=1}^{N}\Big\{&\mathrm{Tr}(P_m R_{u_n} P_l)\\ &\times \frac{c_{n,m}c_{n,l}}{\sigma_{v_n}^2}\lambda^{2j}[A^{j+1}]_{m,k}[A^{j+1}]_{l,k}\Big\}\end{aligned}} \quad (21)$$

where

$$P_m = \left[\frac{1}{1-\lambda}\sum_{r=1}^{N}\frac{c_{r,m}R_{u_r}}{\sigma_{v_r}^2}\right]^{-1}. \quad (22)$$

We now find an expression for the EMSE of the diffusion RLS algorithm (12). This can be done by noting that under assumption 1, $u_{k,i}$ is independent of $\tilde{w}_{k,i-1}$. Then we have

$$\begin{aligned}\mathrm{E}|u_{k,i}\tilde{w}_{k,i-1}|^2 &= \mathrm{E}\mathrm{Tr}(u_{k,i}^* u_{k,i}\tilde{w}_{k,i-1}\tilde{w}_{k,i-1}^*)\\ &= \mathrm{Tr}(R_{u_k}\mathrm{E}\{\tilde{w}_{k,i-1}\tilde{w}_{k,i-1}^*\})\end{aligned}$$

and we obtain

$$\boxed{\begin{aligned}\mathrm{EMSE}_k^{\mathrm{diff}} = \lim_{i\to\infty}\sum_{j=0}^{i}\sum_{l=1}^{N}\sum_{m=1}^{N}\sum_{n=1}^{N}\Big\{&\mathrm{Tr}(R_{u_k}P_m R_{u_n} P_l)\\ &\times \frac{c_{n,m}c_{n,l}}{\sigma_{v_n}^2}\lambda^{2j}[A^{j+1}]_{m,k}[A^{j+1}]_{l,k}\Big\}.\end{aligned}} \quad (23)$$

We summarize our results in the following lemma.

Lemma 1: Under assumptions 1 and 2, the diffusion RLS algorithm (12) is asymptotically unbiased; its mean-square deviation is given by (21), and its excess mean-square error is given by (23).

2) Invariant Spatial Profile: We now specialize (21) and (23) for the case where both the noise variances and the regressor covariance matrices are the same for every node [1]. That is, we assume

$$\begin{aligned}\sigma_{v_1}^2 &= \cdots = \sigma_{v_N}^2 \triangleq \sigma_v^2\\ R_{u_1} &= \cdots = R_{u_N} \triangleq R_u.\end{aligned}$$

In this case, we see that P_m in (22) becomes independent of m and we get

$$P_1 = \cdots = P_N = (1-\lambda)\sigma_v^2 R_u^{-1} \triangleq P. \quad (24)$$

Then we have that

$$\mathrm{Tr}(P_m R_{u_n} P_l)/\sigma_{v_n}^2 = (1-\lambda)^2\sigma_v^2\mathrm{Tr}(R_u^{-1})$$

and (21) becomes

$$\mathrm{MSD}_k^{\mathrm{diff}} = (1-\lambda)^2\sigma_v^2\mathrm{Tr}\left(R_u^{-1}\right)s_{k,k} \quad (25)$$

where

$$s_{k,k} = \lim_{i\to\infty}\sum_{j=0}^{i}\sum_{l=1}^{N}\sum_{m=1}^{N}\sum_{n=1}^{N}\lambda^{2j}c_{n,m}c_{n,l}[A^{j+1}]_{m,k}[A^{j+1}]_{l,k}$$

and is also the $\{k,k\}$ element of the matrix

$$S = \lim_{i\to\infty}\sum_{j=0}^{i}\lambda^{2j}(A^T)^{j+1}C^T C A^{j+1}. \quad (26)$$

This is the same expression we derived in [1], albeit in a different manner. The EMSE in this case is

$$\mathrm{EMSE}_k^{\mathrm{diff}} = (1-\lambda)^2\sigma_v^2 M s_{k,k}. \quad (27)$$

The result can be further simplified for the case $A = A^T = C$. Again using the fact that powers of A have positive elements between 0 and 1, we conclude that the steady state MSD is given by

$$\mathrm{MSD}_k^{\mathrm{diff}} = (1-\lambda)^2\sigma_v^2\mathrm{Tr}\left(R_u^{-1}\right)e_k^T A^4[I-\lambda^2A^2]^{-1}e_k. \quad (28)$$

3) Global Solution: We now show how to specialize (21) and (23) to find the MSD and EMSE of the global least-squares solution (4). This can be accomplished by choosing weighting matrices with constant entries: $A = C = (1/N)\mathbb{1}\mathbb{1}^T$. In this case, we have

$$\sum_{n=1}^{N}\mathrm{Tr}(P_m R_{u_n} P_l)/\sigma_{v_n}^2 = (1-\lambda)^2N^2\mathrm{Tr}\left(\sum_{n=1}^{N}R_{u_n}/\sigma_{v_n}^2\right)^{-1}.$$

Using the fact that the matrix $A = A^2$, we arrive at

$$\mathrm{MSD}_k^{\mathrm{global}} = \frac{1-\lambda}{1+\lambda}\mathrm{Tr}\left(\sum_{r=1}^{N}\frac{R_{u_r}}{\sigma_{v_r}^2}\right)^{-1} \quad (29)$$

and

$$\mathrm{EMSE}_k^{\mathrm{global}} = \frac{1-\lambda}{1+\lambda}\mathrm{Tr}\left[R_{u_k}\left(\sum_{r=1}^{N}\frac{R_{u_r}}{\sigma_{v_r}^2}\right)^{-1}\right]. \quad (30)$$

If we also assume equal noise variances and regressor covariances among nodes, we obtain

$$\mathrm{MSD}_k^{\mathrm{global}} = \frac{1-\lambda}{1+\lambda}\frac{\sigma_v^2}{N}\mathrm{Tr}\left(R_u^{-1}\right)$$

which was also presented in [1]. For the EMSE, we get

$$\mathrm{EMSE}_k^{\mathrm{global}} = \frac{1-\lambda}{1+\lambda}\frac{\sigma_v^2 M}{N}.$$

These are known results for the MSD and EMSE of regular RLS with λ close to 1 [11]. In Section VI we show simulation results for expressions (21), (23), (29), and (30).

4) Limited Cooperation: We now compare the performance of the diffusion RLS algorithm to other cases where cooperation is limited. For simplicity, we assume an invariant spatial profile as before. If we choose $A = C = I$ in (21), we obtain a network where all the nodes are isolated. In this case, the MSD at node k is given by

$$\mathrm{MSD}_k^{\mathrm{isolated}} = \frac{1-\lambda}{1+\lambda}\sigma_v^2\mathrm{Tr}(R_u^{-1}).$$

Another network configuration is obtained by choosing only $C = I$. In this case, nodes only use their own data to update the estimates, but share the estimates in the neighborhood to perform averaging. The MSD becomes

$$\mathrm{MSD}_k^{\text{share}} = \frac{1-\lambda}{1+\lambda}\sigma_v^2 \mathrm{Tr}\left(R_u^{-1}\right)(1-\lambda^2)s_{k,k}$$

where $s_{k,k}$ is the $\{k,k\}$ element of S from (26), using $C = I$. Note that in this case, the expression for the MSD is similar to the one of diffusion RLS (12), though simulations show that this method has slower convergence, and in general a much higher MSD than diffusion RLS (12) when the invariant spatial profile assumption is dropped.

Finally, we may think of a case where nodes exchange data with their neighbors, but do not perform the spatial averaging step. In this case, we obtain the RLS solution using local data, and the MSD is obtained by using $A = I$ in (21) as follows:

$$\mathrm{MSD}_k^{\text{local}} = \frac{1-\lambda}{1+\lambda}\sigma_v^2 \mathrm{Tr}\left(R_u^{-1}\right)\sum_{n=1}^{N} c_{n,k}^2.$$

Note that this MSD is always lower than the one obtained in the isolated case.

V. Combiner Coefficients

We now consider possible choices for the coefficient matrices A and C with individual elements $a_{i,j}$ and $c_{i,j}$ respectively. Recall that $A \in \mathbb{R}^{N\times N}$ and $C \in \mathbb{R}^{N\times N}$ satisfy

$$c_{i,j} \geq 0 \quad \mathbb{1}^T C = \mathbb{1}^T \quad C\mathbb{1} = \mathbb{1} \quad a_{i,j} \geq 0 \mathbb{1}^T A = \mathbb{1}^T. \tag{31}$$

One possible choice of weights A if symmetry is not required, is

$$a_{l,k} = \begin{cases} \deg_l / \sum_{n\in\mathcal{N}_k} \deg_n, & \text{if } l \in \mathcal{N}_k \\ 0, & \text{otherwise} \end{cases} \tag{32}$$

where $\deg_j$ denotes the degree of node j, i.e., the cardinality of its closed neighborhood. A possible choice of weights C is the Metropolis weights, defined by

$$c_{l,k} = \begin{cases} 1/\max\{\deg_l, \deg_k\}, & \text{if } l \in \mathcal{N}_k, l \neq k \\ 1 - \sum_{l\neq k} c_{l,k}, & \text{if } l = k \\ 0, & \text{otherwise.} \end{cases} \tag{33}$$

Note that the Metropolis weights yield a symmetric matrix, and, thus, are appropriate for C. The choices for A and C as in (32) and (33) have been observed to yield good results for the diffusion RLS algorithm (12) in general.

A. Optimization of Matrix A

We can improve the steady state performance of the adaptive network by choosing matrices A and C such that the mean-square deviation is minimized. We therefore consider the following problem:

$$\underset{A,C}{\text{minimize}} \quad \sum_{k=1}^{N} \mathrm{MSD}_k^{\text{diff}} \qquad \text{subject to} \quad (31). \tag{34}$$

This problem is in general not convex. Nevertheless, for the special case of invariant spatial profile, if we fix the matrix C (for example, by using the Metropolis weights (33)), and add the constraint $A = A^T$, (34) becomes a convex optimization problem, and can therefore be solved using standard solvers. We will show that this is the case when we use expression (25). In this case, the problem can be formulated as

$$\underset{A}{\text{minimize}} \quad \lim_{i\to\infty} \mathrm{Tr}\, S_i \quad \text{subject to (31) and } A = A^T \tag{35}$$

where

$$S_i = \sum_{j=0}^{i} \lambda^{2j} (A^T)^{j+1} C^T C A^{j+1}. \tag{36}$$

Now note that

$$S_i - \lambda^2 A S_i A = AC^T CA - \lambda^{2(i+1)} A^{i+2} C^T C A^{i+2}$$

and also

$$[I - \lambda A] S_i [I + \lambda A] = S_i - \lambda A S_i + \lambda S_i A - \lambda^2 A S_i A.$$

Combining these two results, and taking the trace, we arrive at

$$\begin{aligned}\mathrm{Tr}(S_i) = {} & \lambda \mathrm{Tr}\left[(I-\lambda^2 A^2)^{-1}(S_i A - A S_i)\right] \\ & + \mathrm{Tr}\left[(I-\lambda^2A^2)^{-1}\right. \\ & \left. \times (AC^TCA - \lambda^{2(i+1)} A^{i+2} C^T C A^{i+2})\right].\end{aligned}$$

Since A and S_i are symmetric, and using the result $\mathrm{Tr}(X) = \mathrm{Tr}(X^T)$, the first term of the above equation is zero. Taking the limit when i goes to infinity, we obtain

$$\begin{aligned}\lim_{i\to\infty} \mathrm{Tr}(S_i) &= \mathrm{Tr}\left[CA(I-\lambda^2A^2)^{-1}AC^T\right] \\ &= -\lambda^{-2}\mathrm{Tr}[C^TC((I-\lambda^2A^2)^{-1} \\ &\qquad - (I+\lambda A)^{-1} - (I-\lambda A)^{-1} + I)].\end{aligned}$$

Using the identity

$$[(I-B)(I+B)]^{-1} = [(I+B)^{-1} + (I-B)^{-1}]/2$$

we arrive at

$$\lim_{i\to\infty} \mathrm{Tr}(S_i) = \frac{1}{2\lambda^2}\mathrm{Tr}\{C^TC[(I+\lambda A)^{-1} + (I-\lambda A)^{-1} - I]\}.$$

This expression is convex in the elements of A, since for positive definite X, $\mathrm{Tr}(X^{-1})$ is convex in the elements of X [19] and so is $\mathrm{Tr}(YX^{-1})$ for any constant matrix Y. Also, A is related linearly to X, and linear transformations preserve convexity. Thus, since the constraints (31) are linear, problem (35) is a convex problem, and can be solved using a convex optimization solver.

Unfortunately, the restriction that A be a symmetric matrix for the problem to be convex does not always produce good results. For instance, a choice of A as in (32) may give a lower MSD than the optimal symmetric matrix A. This observation

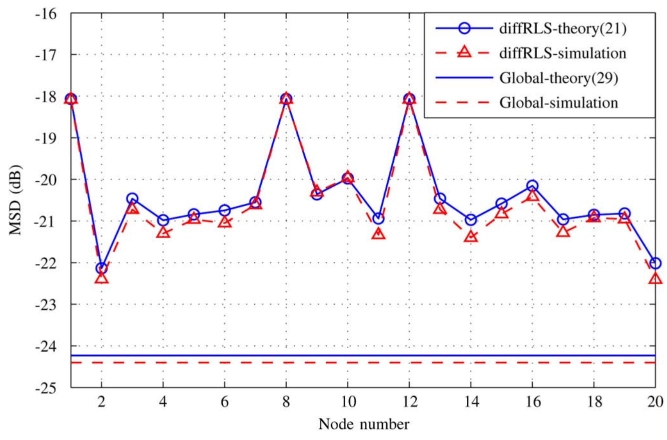

Fig. 3. Steady-state MSD values for diffusion RLS (12) and the global solution (4), using $\lambda = 0.9$.

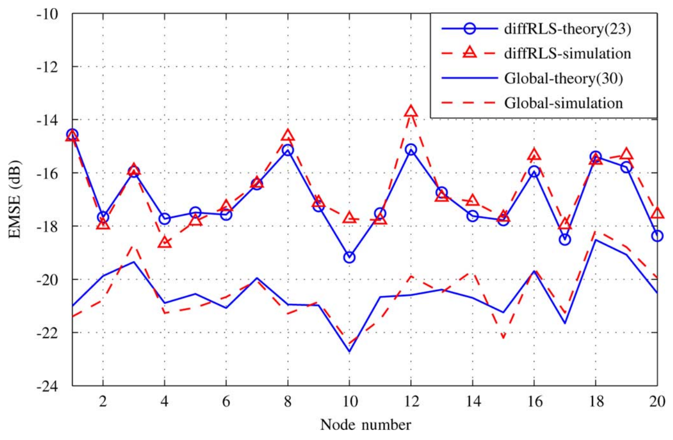

Fig. 4. Steady-state EMSE values for diffusion RLS (12) and the global solution (4), using $\lambda = 0.9$.

is highly dependent on the configuration of the network. For example, when the configuration is such that a choice like (32) produces a symmetric matrix, then the optimized weights will show improvement.

B. Optimization of Matrix C

We can also fix matrix A, and solve problem (34) by minimizing with respect to C. This is also a convex problem in the elements of C, as follows. Let $X_j = CA^{j+1}$. From (36) we have

$$\text{Tr}S_i = \sum_{j=0}^{i} \lambda^{2j} \text{Tr}\left(X_j^T X_j\right) = \sum_{j=0}^{i} \lambda^{2j} \sum_{m=1}^{N} \sum_{n=1}^{N} [X_j]_{m,n}^2.$$

The previous expression is a sum of convex functions of the elements of X_j. Since the elements of X_j are related linearly to the elements of C, and since linear transformations preserve convexity, we have that problem (34) is convex in the elements of C.

VI. SIMULATIONS

We now show simulation results for the diffusion RLS algorithm (12), and compare its performance to the theoretical results of Section IV and also to other distributed algorithms.

The measurements were generated according to model (1), and the regressors $u_{k,i}$ were chosen Gaussian iid. The network had a total of $N = 20$ nodes and the size of the unknown vector w^o was $M = 5$. The results were averaged over 200 experiments. For diffusion RLS, unless otherwise indicated, Metropolis weights were used for C, and A was chosen as in (32).

Fig. 3 shows the MSD results obtained from simulations and from expression (21) for diffusion RLS (12) and (29) for the global solution (4), for a value of $\lambda = 0.9$. The noise variances were chosen different for every node, and the regressor covariance matrices were generated at random. Fig. 4 shows the respective plots for the EMSE. We see that the theoretical results agree well with the simulations.

Fig. 5 compares the proposed diffusion RLS algorithm (12) with the distributed RLS (dRLS) of [3], [4] and the space-time diffusion (STD) algorithm of [10]. To simulate STD, Metropolis

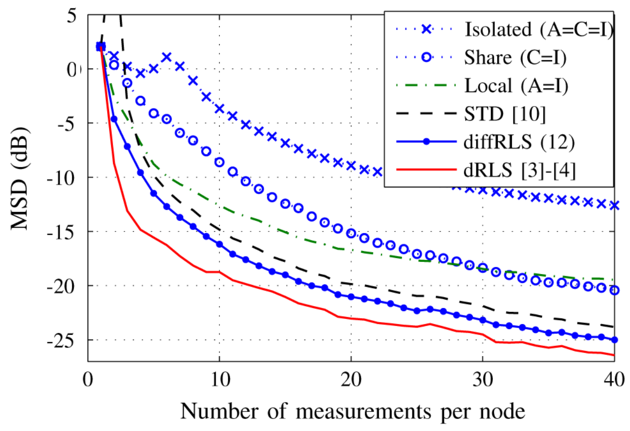

Fig. 5. MSD curves of different algorithms, using $\lambda = 1$.

weights were used as proposed in [10]. Since STD does not account for the forgetting factor λ, we set $\lambda = 1$ for diffusion RLS in the simulations in order to ensure a fair comparison. We observe that diffusion RLS (12) has better convergence performance than STD, even though the latter requires transmission of more information and inversion of matrices for every measurement (see Appendix B for a discussion on this observation). Also, as indicated in [3], [4], the dRLS solution achieves the global LS performance at the cost of requiring a cyclic path. Also shown are the special cases discussed in Section IV-C-4) where cooperation is limited. The curve labeled "Isolated" represents the case $A = C = I$ where the nodes are isolated, the curve labeled "Share" corresponds to the case $C = I$, and the curve labeled "Local" corresponds to the case $A = I$. As expected, all these latter cases with limited cooperation are inferior in performance to diffusion RLS (12).

Fig. 6 shows the MSD of different algorithms as a function of the number of scalars communicated per node. We observe that diffusion RLS exhibits significant improvement over STD. That is, in order to obtain a fixed MSD, diffusion RLS requires less communications (measured in terms of scalars transmitted) than STD, and the performance in this case is close to that of dRLS.

The algorithms were also compared using different data models, where the time independence and identically distributed

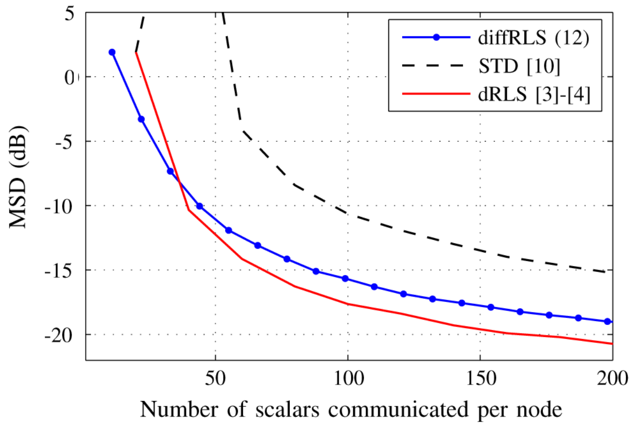

Fig. 6. MSD curves of different algorithms as a function of number of scalars communicated per node, using $\lambda = 1$.

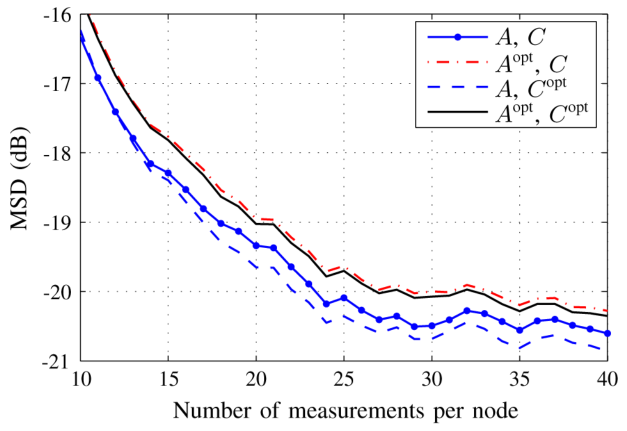

Fig. 8. MSD curves for different choices of weighting matrices A and C.

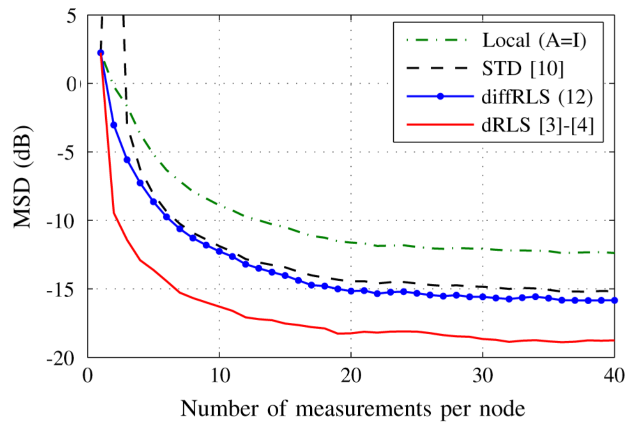

Fig. 7. MSD curves of different algorithms, using $\lambda = 1$ and relaxed data models.

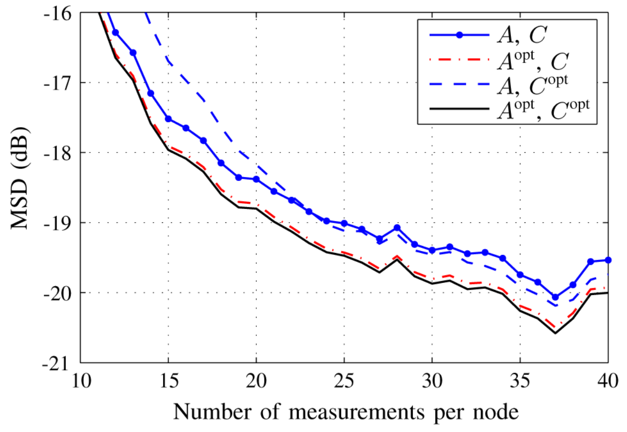

Fig. 9. MSD curves for different choices of weighting matrices A and C.

assumptions were dropped. Fig. 7 shows the performance of the algorithms when the noise variances were different for every node (between 0.1 and 0.2), and the regressors were chosen to have shift structure with the elements of $u_{k,i}$ forming a Markov process with different coefficients for every node. Again a value of $\lambda = 1$ was used. We did not observe a significant degradation in performance by using different data models.

Finally, we illustrate the effect of the coefficient optimization procedure of Section V on performance. In Fig. 8, we show the MSD curves for the diffusion RLS algorithm (12), using four different sets of coefficients and $\lambda = 0.9$. The choices A and C represent the matrices from (32) and (33), respectively. The choices A^{opt} and C^{opt} represent the matrices obtained by solving the convex optimization problems of Section V such as 35. We clearly observe that when we use A^{opt} instead of A, performance is degraded. The reason is that A^{opt} offers the best performance over all *symmetric* matrices, but A has better performance. We can also observe that the optimization of C for a fixed A always improves the results.

In Fig. 9 we present the same curves, but for a different network where A in (32) happens to be symmetric. Now the optimization of A is clearly improving performance, as expected, and so is the optimization of C. The best performance is obtained when both matrices are optimized.

VII. Conclusions and Future Work

We have addressed the problem of distributed estimation over adaptive networks. We have proposed a diffusion RLS algorithm that obtains good performance compared to the global solution, and outperforms previous methods. Furthermore, it does not require transmission or inversion of matrices, and has no topology constraints. We have shown mean and mean-squared convergence under ergodicity assumptions and derived expressions for the mean-square deviation and excess mean-square error, which agree well with the simulation results. The algorithm was simulated under some general conditions and exhibited good performance. Finally, we described how to choose the weighting matrices by solving appropriate convex optimization problems.

A useful question to pursue is the study and development of diffusion variants that involve coarse quantization in order to minimize the number of bits communicated between nodes. Interesting work in this regard using 1-bit communications has been proposed in the works [20], [21] in the context of other distributed procedures. Also of importance is the issue of reliability of the measurements obtained by the nodes. These topics will be addressed in future work.

APPENDIX A
DERIVATION OF THE DIFFUSION RLS ALGORITHM (12)

The solution of (5) can be obtained from (3) by replacing $\mathcal{W}_i$ with $\mathcal{W}_{k,i}$ as in (6), $\Pi_i = \lambda^{i+1}\Pi$ and $\bar{w} = 0$. It is given by

$$\psi_{k,i} = P_{k,i}\mathcal{H}_i^*\mathcal{W}_{k,i}\mathcal{Y}_i \tag{37}$$

where

$$P_{k,i} = \left[\lambda^{i+1}\Pi + \mathcal{H}_i^*\mathcal{W}_{k,i}\mathcal{H}_i\right]^{-1}.$$

We are interested in a recursion that allows us to compute $P_{k,i}$ from $P_{k,i-1}$. We note that

$$\begin{aligned} P_{k,i}^{-1} &= \lambda\left[\lambda^i\Pi + \mathcal{H}_{i-1}^*\mathcal{W}_{k,i-1}\mathcal{H}_{i-1}\right] + H_i^*R_v^{-1}C_kH_i \\ &= \lambda P_{k,i-1}^{-1} + H_i^*R_v^{-1}C_kH_i \\ &= \lambda P_{k,i-1}^{-1} + \sum_{l=1}^{N}\frac{c_{l,k}}{\sigma_{v_l}^2}u_{l,i}^*u_{l,i}. \end{aligned}$$

We can obtain $P_{k,i}$ from $P_{k,i-1}$ using a series of rank-one updates as follows:

$$\begin{aligned} &P_{k,i}^0 \leftarrow \lambda^{-1}P_{k,i-1} \\ &\text{For } l = 1 \text{ to } N, \text{repeat}: \\ &\qquad P_{k,i}^l \leftarrow \left[\left(P_{k,i}^{l-1}\right)^{-1} + \frac{c_{l,k}}{\sigma_{v_l}^2}u_{l,i}^*u_{l,i}\right]^{-1} \\ &\text{end} \\ &P_{k,i} \leftarrow P_{k,i}^N \end{aligned}$$

where we have introduced the matrices $P_{k,i}^l$ to denote the intermediate results after every rank-one update. Using the matrix inversion lemma, and noting that we need only consider values of l where $c_{l,k} \neq 0$, the above recursions become

$$\begin{aligned} &P_{k,i} \leftarrow \lambda^{-1}P_{k,i-1} \\ &\text{For every } l \in \mathcal{N}_k, \text{ repeat}: \\ &\qquad P_{k,i} \leftarrow P_{k,i} - \frac{c_{l,k}P_{k,i}u_{l,i}^*u_{l,i}P_{k,i}}{\sigma_{v_l}^2 + c_{l,k}u_{l,i}P_{k,i}u_{l,i}^*} \\ &\text{end.} \end{aligned} \tag{38}$$

We now need a recursion for the update of the estimate $\psi_{k,i}$. Define the matrices $\mathcal{Y}_i^l$ and $\mathcal{H}_i^l$ that collect all measurements and regressors, respectively, from all nodes up to time $i-1$, and from nodes 1 to l at time i. The matrices are given by

$$\mathcal{Y}_i^l = \begin{bmatrix} y_l \\ \vdots \\ y_1 \\ \hline \mathcal{Y}_{i-1} \end{bmatrix} \text{ and } \mathcal{H}_i^l = \begin{bmatrix} u_{l,i} \\ \vdots \\ u_{1,i} \\ \hline \mathcal{H}_{i-1} \end{bmatrix}.$$

Also define the diagonal weight matrix $\mathcal{W}_{k,i}^l$ as

$$\mathcal{W}_{k,i}^l = \begin{bmatrix} c_{l,k}/\sigma_{v_l}^2 & & & \\ & \ddots & & \\ & & c_{1,k}/\sigma_{v_1}^2 & \\ & & & \lambda\mathcal{W}_{k,i-1} \end{bmatrix}.$$

Consider the intermediate estimates

$$\psi_{k,i}^l = P_{k,i}^l\left(\mathcal{H}_i^l\right)^*\mathcal{W}_{k,i}^l\mathcal{Y}_i^l$$

where

$$P_{k,i}^l = \left[\lambda^{i+1}\Pi + (\mathcal{H}_i^l)^*\mathcal{W}_{k,i}^l\mathcal{H}_i^l\right]^{-1}.$$

It holds that $\psi_{k,i-1} = \psi_{k,i-1}^N \overset{\triangle}{=} \psi_{k,i}^0$. Thus, we can calculate $\psi_{k,i}$ recursively from $\psi_{k,i-1}$ by instead calculating $\psi_{k,i}^N$ recursively from $\psi_{k,i}^0$. Let j denote the *smallest* index such that $c_{j,k} \neq 0$. We calculate $\psi_{k,i}^j$ from $\psi_{k,i}^0$ as follows:

$$\begin{aligned} \psi_{k,i}^j &= P_{k,i}^j\left[\lambda(\mathcal{H}_i^0)^*\mathcal{W}_{k,i}^0\mathcal{Y}_i^0 + \frac{c_{j,k}}{\sigma_{v_j}^2}u_{j,i}^*d_j(i)\right] \\ &= \underbrace{\lambda P_{k,i}^0(\mathcal{H}_i^0)^*\mathcal{W}_{k,i}^0\mathcal{Y}_i^0}_{\psi_{k,i-1}} \\ &\quad + \frac{c_{j,k}}{\sigma_{v_j}^2}P_{k,i}^0u_{j,i}^*\left(1 - \frac{c_{j,k}u_{j,i}P_{k,i}^0u_{j,i}^*}{\sigma_{v_j}^2 + c_{j,k}u_{j,i}P_{k,i}^0u_{j,i}^*}\right)d_j(i) \\ &\quad - \frac{c_{j,k}P_{k,i}^0u_{j,i}^*u_{j,i}}{\sigma_{v_j}^2 + c_{j,k}u_{j,i}P_{k,i}^0u_{j,i}^*}\underbrace{\lambda P_{k,i}^0(\mathcal{H}_i^0)^*\mathcal{W}_{k,i}^0\mathcal{Y}_i^0}_{\psi_{k,i-1}} \\ &= \psi_{k,i-1} + \frac{c_{j,k}P_{k,i}^0u_{j,i}^*}{\sigma_{v_j}^2 + c_{j,k}u_{j,i}P_{k,i}^0u_{j,i}^*}(d_j(i) \\ &\quad - u_{j,i}\psi_{k,i-1}) \end{aligned} \tag{39}$$

In a similar fashion, we can derive recursions to obtain $\psi_{k,i}^l$ from $\psi_{k,i}^{l-1}$, for $l = j+1,\ldots,N$. We now have

$$\begin{aligned} \psi_{k,i}^l &= P_{k,i}^l\left[(\mathcal{H}_i^{l-1})^*\mathcal{W}_{k,i}^{l-1}\mathcal{Y}_i^{l-1} + \frac{c_{l,k}}{\sigma_{v_l}^2}u_{l,i}^*d_l(i)\right] \\ &= \psi_{k,i-1}^{l-1} + \frac{c_{l,k}P_{k,i}^{l-1}u_{l,i}^*}{\sigma_{v_l}^2 + c_{l,k}u_{l,i}P_{k,i}^{l-1}u_{l,i}^*}(d_l(i) \\ &\quad - u_{l,i}\psi_{k,i-1}^{l-1}). \end{aligned} \tag{40}$$

We can combine (38), (39) and (40) into a single recursion as follows:

$$\begin{aligned} &\psi_{k,i}^0 \leftarrow w_{k,i-1} \\ &P_{k,i}^0 \leftarrow \lambda^{-1}P_{k,i-1} \\ &\text{for } l = 1 \text{ to } N, \text{ repeat} \\ &\qquad \psi_{k,i}^l \leftarrow \psi_{k,i}^{l-1} + \frac{c_{l,k}P_{k,i}^{l-1}u_{l,i}^*[d_l(i) - u_{l,i}\psi_{k,i}]}{\sigma_{v_l}^2 + c_{l,k}u_{l,i}P_{k,i}^{l-1}u_{l,i}^*} \\ &\qquad P_{k,i}^l \leftarrow P_{k,i}^{l-1} - \frac{c_{l,k}P_{k,i}^{l-1}u_{l,i}^*u_{l,i}P_{k,i}^{l-1}}{\sigma_{v_l}^2 + c_{l,k}u_{l,i}P_{k,i}^{l-1}u_{l,i}^*} \\ &\text{end} \\ &P_{k,i} \leftarrow P_{k,i}^N \\ &\psi_{k,i} \leftarrow \psi_{k,i}^N. \end{aligned}$$

To simplify notation, we drop the super-indexes j in the above recursions, and note that we only need to consider values of l for which $c_{l,k} \neq 0$. Considering all time instants up to i, problem (5) can be solved recursively via the following algorithm:

```
Start with $w_{k,-1} = 0$ and $P_{k,-1} = \Pi^{-1}$ for each node $k$.
For every time instant $i$, and for every node $k$, repeat
  $\psi_{k,i} = w_{k,i-1}$
  $P_{k,i} = \lambda^{-1} P_{k,i-1}$
  for all $l \in \mathcal{N}_k$
    $\psi_{k,i} \leftarrow \psi_{k,i} + \frac{c_{l,k} P_{k,i} u_{l,i}^* [d_l(i) - u_{l,i}\psi_{k,i}]}{\sigma_{v_l}^2 + c_{l,k} u_{l,i} P_{k,i} u_{l,i}^*}$
    $P_{k,i} \leftarrow P_{k,i} - \frac{c_{l,k} P_{k,i} u_{l,i}^* u_{l,i} P_{k,i}}{\sigma_{v_l}^2 + c_{l,k} u_{l,i} P_{k,i} u_{l,i}^*}$
end
$w_{k,i} = \psi_{k,i}$.                                              (41)
```

We have referred to this recursions in Section VI as the "Local" solution, since nodes only take into account data from their neighborhoods to compute the estimates.

The diffusion RLS algorithm takes further advantage of the interconnection between nodes to improve the estimation by also exchanging estimates. As argued in Section II, we replace the last step in (41) by a weighted average of the estimates of the nodes as in (11). This immediately leads to (12).

Appendix B
Relation Between Diffusion RLS (12) and STD [10]

The space-time diffusion (STD) algorithm of [10] obtains an estimate $w_{k,i}$ of w^o for node k at time i through the following relations:

$$P_{k,i}^{-1} = \sum_{l=1}^{N} c_{l,k} \left[P_{l,i-1}^{-1} + u_{l,i}^* u_{l,i} / \sigma_{v_l}^2 \right]$$
$$q_{k,i} = \sum_{l=1}^{N} c_{l,k} \left[q_{l,i-1} + u_{l,i}^* d_l(i) / \sigma_{v_l}^2 \right]$$
$$w_{k,i} = P_{k,i} q_{k,i}.$$

Consider now the following approximations:

$$\sum_{l=1}^{N} c_{l,k} P_{l,i}^{-1} \approx P_{k,i}^{-1} \tag{42}$$

and

$$\sum_{l=1}^{N} c_{l,k} q_{l,i} = \sum_{l=1}^{N} c_{l,k} P_{l,i}^{-1} w_{l,i} \approx P_{k,i}^{-1} \sum_{l=1}^{N} c_{l,k} w_{l,i}. \tag{43}$$

The above approximations are reasonable for large i, under assumption 2 and when we have an invariant spatial profile. Then we get

$$P_{k,i} \approx \left[P_{k,i-1}^{-1} + \sum_{l=1}^{N} c_{l,k} u_{l,i}^* u_{l,i} / \sigma_{v_l}^2 \right]^{-1} \tag{44}$$

$$q_{k,i} \approx P_{k,i-1}^{-1} \bar{w}_{k,i-1} + \sum_{l=1}^{N} c_{l,k} u_{l,i}^* d_l(i) / \sigma_{v_l}^2 \tag{45}$$

$$w_{k,i} = P_{k,i} q_{k,i} \tag{46}$$

where $\bar{w}_{k,i-1} = \sum_l c_{l,k} w_{l,i-1}$. Equations (44)–(46) produce the same result as the one we would obtain after the incremental update of diffusion RLS (12), provided $\lambda = 1$ and $A = C$. Thus, one expects STD and the diffusion RLS algorithm (12) to give similar results when approximations (42)–(43) hold. Nevertheless, our experience shows that diffusion RLS outperforms STD in general since it does not require $A = C$, and also allows for the use of a forgetting factor λ.

References

[1] F. S. Cattivelli, C. G. Lopes, and A. H. Sayed, "A diffusion RLS scheme for distributed estimation over adaptive networks," in *Proc. IEEE Workshop Signal Process. Advances Wireless Commun. (SPAWC)*, Helsinki, Finland, Jun. 2007, pp. 1–5.

[2] D. Estrin, L. Girod, G. Pottie, and M. Srivastava, "Instrumenting the world with wireless sensor networks," in *Proc. Int. Conf. Acoustics, Speech, Signal Process. (ICASSP)*, Salt Lake City, UT, May 2001, vol. 4, pp. 2033–2036.

[3] A. H. Sayed and C. G. Lopes, "Distributed recursive least-squares strategies over adaptive networks," in *Proc. 40th Asilomar Conf. Signals, Syst. Comput.*, Pacific Grove, CA, Oct. 2006, pp. 233–237.

[4] A. H. Sayed and C. G. Lopes, "Adaptive processing over distributed networks," *IEICE Trans. Fund. Electron., Communi. Comput. Sci.*, vol. E90-A, no. 8, pp. 1504–1510, Aug. 2007.

[5] C. G. Lopes and A. H. Sayed, "Incremental adaptive strategies over distributed networks," *IEEE Trans. Signal Process.*, vol. 55, no. 8, pp. 4064–4077, Aug. 2007.

[6] C. G. Lopes and A. H. Sayed, "Diffusion least-mean squares over adaptive networks," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Honolulu, HI, Apr. 2007, pp. 917–920.

[7] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," in *Proc. Int. Conf. Information Processing Sensor Networks (IPSN)*, Los Angeles, CA, Apr. 2005, pp. 63–70.

[8] R. Olfati-Saber and J. S. Shamma, "Consensus filters for sensor networks and distributed sensor fusion," in *Proc. 44th IEEE Conf. Decision Contr.*, Dec. 2005, pp. 6698–6703.

[9] V. D. Blondel, J. M. Hendrickx, A. Olshevsky, and J. N. Tsitsiklis, "Convergence in multiagent coordination, consensus, and flocking," in *Proc. Joint 44th IEEE Conf. Decision Contr. Europ. Contr. Conf. (CDC-ECC)*, Seville, Spain, Dec. 2005, pp. 2996–3000.

[10] L. Xiao, S. Boyd, and S. Lall, "A space-time diffusion scheme for peer-to-peer least-squares estimation," in *Proc. IPSN*, Nashville, TN, Apr. 2006, pp. 168–176.

[11] A. H. Sayed, *Fundamentals of Adaptive Filtering*. Hoboken, NJ: Wiley, 2003.

[12] J. Arenas-Garcia, A. R. Figueiras-Vidal, and A. H. Sayed, "Mean-square performance of a convex combination of two adaptive filters," *IEEE Trans. Signal Process.*, vol. 54, no. 3, pp. 1078–1090, Mar. 2006.

[13] M. Silva and V. H. Nascimento, "Convex combination of adaptive filters with different tracking capabilities," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Honolulu, HI, Apr. 2007, vol. 3, pp. 925–928.

[14] Y. Zhang and J. A. Chambers, "Convex combination of adaptive filters for a variable tap-length LMS algorithm," *IEEE Signal Process. Lett.*, vol. 13, no. 10, pp. 628–631, Oct. 2006.

[15] S. Haykin, *Adaptive Filter Theory*. Englewood Cliffs, NJ: Prentice-Hall, 2001.

[16] B. Widrow and S. Stearns, *Adaptive Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1985.

[17] N. Kalouptsidis and S. Theodoridis, *Adaptive System Identification and Signal Processing Algorithms*. Englewood Cliffs, NJ: Prentice-Hall, 1993.

[18] H. Sakai, J. M. Yang, and T. Oka, "Exact convergence analysis of adaptive filter algorithms without the persistently exciting condition," *IEEE Trans. Signal Process.*, vol. 55, no. 5, pp. 2077–2083, May 2007.

[19] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[20] A. Ribeiro and G. B. Giannakis, "Bandwidth-constrained distributed estimation for wireless sensor networks—Part I: Gaussian case," *IEEE Trans. Signal Process.*, vol. 54, no. 3, pp. 1131–1143, Mar. 2006.

[21] J.-J. Xiao, Z.-Q. Luo, and G. B. Giannakis, "Performance bounds for the rate-constrained universal decentralized estimators," *IEEE Signal Process. Lett.*, vol. 14, no. 1, pp. 47–50, 2007.

Federico S. Cattivelli (S'03) received the B.S. degree in electronics engineering from Universidad ORT, Uruguay, in 2004 and the M.S. degree from the University of California, Los Angeles (UCLA) in 2006. He is currently working towards the Ph.D. degree in electrical engineering at UCLA. He is a member of the Adaptive Systems Laboratory, UCLA, where he conducts his research under the supervision of Prof. A. H. Sayed.

His research interests are in signal processing, adaptive filtering, estimation theory, and optimization. His current research activities are focused in the areas of distributed estimation over adaptive networks, Doppler shift frequency estimation for space applications (in collaboration with the NASA Jet Propulsion Laboratory), and state estimation and modeling in biomedical systems (in collaboration with the UCLA School of Medicine).

Mr. Cattivelli was a recipient of the UCLA Graduate Division and UCLA Henry Samueli SEAS Fellowships in 2005 and 2006.

Cassio G. Lopes (S'06) received the B.S. and M.S. degrees in electrical engineering from the Federal University of Santa Catarina, Brazil, in 1997 and 1999, respectively, and the M.S. degree in electrical engineering from the University of California, Los Angeles (UCLA), in 2004. He is currently working towards the Ph.D. degree in electrical engineering at the Adaptive Systems Laboratory (ASL/UCLA), in distributed adaptive estimation.

From 1999 to 2002, he worked with hardware design for biomedical signals acquisition and transmission over the Internet, and devised simple image compressing schemes. He also taught undergraduate-level courses in computer science and electrical engineering. Since 2005, he has been developing robust and efficient frequency trackers for direct-to-Earth Mars communications, in cooperation with the NASA Jet Propulsion Laboratory. His current research interests are theory and methods for adaptive and statistical signal processing, distributed adaptive algorithms, as well as frequency tracking for space communications.

Ali H. Sayed (F'01) received the Ph.D. degree in electrical engineering from Stanford University, Stanford, CA, in 1992.

He is Professor and Chairman of electrical engineering with the University of California, Los Angeles (UCLA). He is also the Principal Investigator of the UCLA Adaptive Systems Laboratory (www.ee.ucla.edu/asl). He has more than 270 journal and conference publications, is the author of *Fundamentals of Adaptive Filtering* (New York: Wiley, 2003), the coauthor of the research monograph *Indefinite Quadratic Estimation and Control* (Philadelphia, PA: SIAM, 1999) and of the graduate-level textbook *Linear Estimation* (Englewood Cliffs, NJ: Prentice-Hall, 2000). He is also coeditor of the volume *Fast Reliable Algorithms for Matrices with Structure* (Philadelphia, PA: SIAM, 1999). He has contributed several articles to engineering and mathematical encyclopedias and handbooks. His research interests span several areas, including adaptive and statistical signal processing, filtering and estimation theories, signal processing for communications, interplays between signal processing and control methodologies, system theory, and fast algorithms for large-scale problems.

Dr. Sayed has served on the program committees of several international meetings. He is the recipient of the 1996 IEEE Donald G. Fink Award. He received the Best Paper Award from the IEEE Signal Processing Society in 2002, the Kuwait Prize in Basic Science in 2003, the Frederick E. Terman Award in 2005, a Young Author Best Paper Award in 2005 from the IEEE Signal Processing Society, and two Best Student Paper awards at international meetings. He is also a member of the technical committees on Signal Processing Theory and Methods (SPTM) and on Signal Processing for Communications (SPCOM), both of the IEEE Signal Processing Society. He has served as Editor-in-Chief of the IEEE TRANSACTIONS ON SIGNAL PROCESSING (2003–2005) and is now serving as Editor-in-Chief of the EURASIP *Journal on Applied Signal Processing*. He is serving as General Chairman of ICASSP 2008 and sits on the Board of Governors of the IEEE Signal Processing Society.